Appendix F
Spatial Data Standard for Facilities, Infrastructure, and Environment and Facility Management Standard for Facilities, Infrastructure, and Environment

F-1. Compliance Policy

a. Since 1993 the Spatial Data Standard for Facilities, Infrastructure, and Environment (SDSFIE) (formerly called TSSDS and SDS) and Facility Management Standard for Facilities, Infrastructure, and Environment (FMSFIE) (formerly called TSFMS and FMS) have continued to expand to meet the needs of an increasing customer community. From Release 1.600 to present, the number of "features" (entity types) has grown from just over 600 to more than 1,000 and the number of "attributes" has increased from less than 10,000 to nearly 26,000. For Release 2.000, an unpopulated database created from the entire SDSFIE/FMSFIE schema will require about 75 megabytes of hard drive space.

b. Due to its generic nature, many SDSFIE/FMSFIE tables contain a large number of attributes. Many customers have indicated they prefer to maintain only the SDSFIE/FMSFIE attributes that they need, and prefer to be able to include their "local" attributes (i.e., the attributes not currently included in the SDSFIE/FMSFIE) within the SDSFIE/FMSFIE attribute tables.

c. In 1997 the concept of a Filter was introduced to provide customers the capability to build and manage less than the full SDSFIE/FMSFIE schema (i.e., the entire SDSFIE/FMSFIE data model/data set). The Filter capability (as provided by the Filter Maker software application) permits customers to select subsets of the SDSFIE/FMSFIE schema for their specific functions and projects.

d. Up to the present time, the policy for SDSFIE/FMSFIE compliance has dictated that if an SDSFIE/FMSFIE attribute table (i.e., an attribute table included in the SDSFIE/FMSFIE schema) is present in the database, then all SDSFIE/FMSFIE attributes (i.e., attributes included in each specific SDSFIE/FMSFIE attribute table) must be present in that table. All attributes in the table must also use the formatting, ordering, and naming conventions of the SDSFIE/FMSFIE. The addition of customer-defined attributes is not allowed in a SDSFIE/FMSFIE table.

e. The SDSFIE/FMSFIE SQL Generator, Access Builder, and GeoMedia Builder tools are designed to build, test, and upgrade databases that meet the Basic level of SDSFIE/FMSFIE compliance. The "Basic" level of SDSFIE/FMSFIE compliance would be recommended for the following:

- Initial Geographic Information Systems (GIS) implementations.
- Customers who do not have experienced database administrators and GIS analysts.
- Where local policies and standards have not yet been developed for ensuring quality and integrity of data collected and delivered by multiple sources.

f. The Experienced level of SDSFIE/FMSFIE compliance would be recommended for customers who have experienced database administrators and/or GIS analysts; and where local policies and standards have been developed for ensuring quality and integrity of data collected and delivered by multiple sources.

g. The SDSFIE and FMSFIE are designed to be used with an external relational database (i.e., Oracle, Informix, Access, etc.) rather than the flat-file data structure; e.g., Environmental Systems Research Institute (ESRI) coverage and shape files and Computer-Aided Design and Drafting (CADD) objects. The use of an external relational database can provide many benefits, such as the following:

- Provide the ability to support multiple users accessing the data simultaneously using different CADD, GIS, and other software products.

- Provide an enterprise GIS solution, where data maintained in one database format (by one or multiple offices) is available to all users at an installation or within an organization.

- Provide a nonproprietary format with greater flexibility to share data with other users and applications.

- Provide a stable data format that protects an organization's data investment (vendors may discontinue products and/or support of a proprietary data structure at some time in the future).

- Avoid possible loss of data when upgrading to newer versions of CADD/GIS software, or exporting graphic and nongraphic data to other CADD/GIS software products/applications

- Maximize the return on investment by providing the capability to use data in an organization's business functions (e.g., generating reports, statistical analyses, etc.) using database, programming, and Web tools.

However, if GIS users do not yet have the experience or resources to establish and maintain an external relational database for GIS, they should use the SDSFIE and FMSFIE as data dictionaries (i.e., use SDSFIE/FMSFIE attribute and domain value naming conventions, data types, etc.). Using the SDSFIE and FMSFIE as data dictionaries for flat-file data structures can simplify and save costs on future conversion of the flat files to an SDSFIE/FMSFIE-compliant relational database.

F-2. Basic Level

Basic level is supported by SDSFIE/FMSFIE SQL Generator, Access Builder, and GeoMedia Builder tools.

a. Entity types and entities. Use SDSFIE entity type and entity naming conventions (i.e., the naming conventions included in the SDSFIE). The SDSFIE symbology (i.e., colors, line styles, or symbols) is recommended, and customers have the option of using their own symbology.

b. Attribute tables. Use SDSFIE/FMSFIE attribute table naming conventions and definitions (short names of eight characters in length). (The SQL Generator and Access Builder software applications currently construct attribute tables using the SDSFIE/FMSFIE short names.) Use of the entire SDSFIE/FMSFIE schema is not a requirement. A subset of the SDSFIE/FMSFIE schema may be used by: selecting one of the custom Filters provided with each SDSFIE/FMSFIE release; or using the SDSFIE/FMSFIE Filter Maker software application to build a custom filter based on the specific entity types required. The Filters will provide a SDSFIE/FMSFIE schema containing all of the selected graphic attribute tables (i.e., those attribute tables associated with the selected entity types) as well as all linked/joined SDSFIE/FMSFIE attribute tables. The unneeded linked/joined SDSFIE/FMSFIE attribute tables may be deleted.

c. Attributes.

(1) Use defined SDSFIE/FMSFIE attribute naming conventions and definitions (short names of 10 characters in length). (The SQL Generator and Access Builder software applications currently construct tables using the SDSFIE/FMSFIE short names.) All SDSFIE/FMSFIE-defined attributes, as output by the SDSFIE/FMSFIE software tools (e.g., SQL Generator and Access Builder), must be kept for the selected tables. Some options, which permit users to exclude specific predefined attributes, are being built into the software tools.

(2) If a locally configured attribute is required, determine the SDSFIE/FMSFIE table with which the attribute is associated. Determine the primary key of that table. This information is easily obtained from the SDSFIE/FMSFIE browser software application. In the database, create a new table. Name this table using the SDSFIE/FMSFIE table name, followed by a locally assigned prefix. As an example, "local_" would work quite well. Thus, if an attribute were to be added to the "bggenstr" SDSFIE/FMSFIE attribute table, a "local_bggenstr" table would be created. The very first attribute in the table would be "buildng_id," the very same attribute that is the primary key of the bggenstr table.

(3) Figure F-1 below shows a graphical depiction of the arrangement being recommended. The two tables parallel each other inside the user's database.

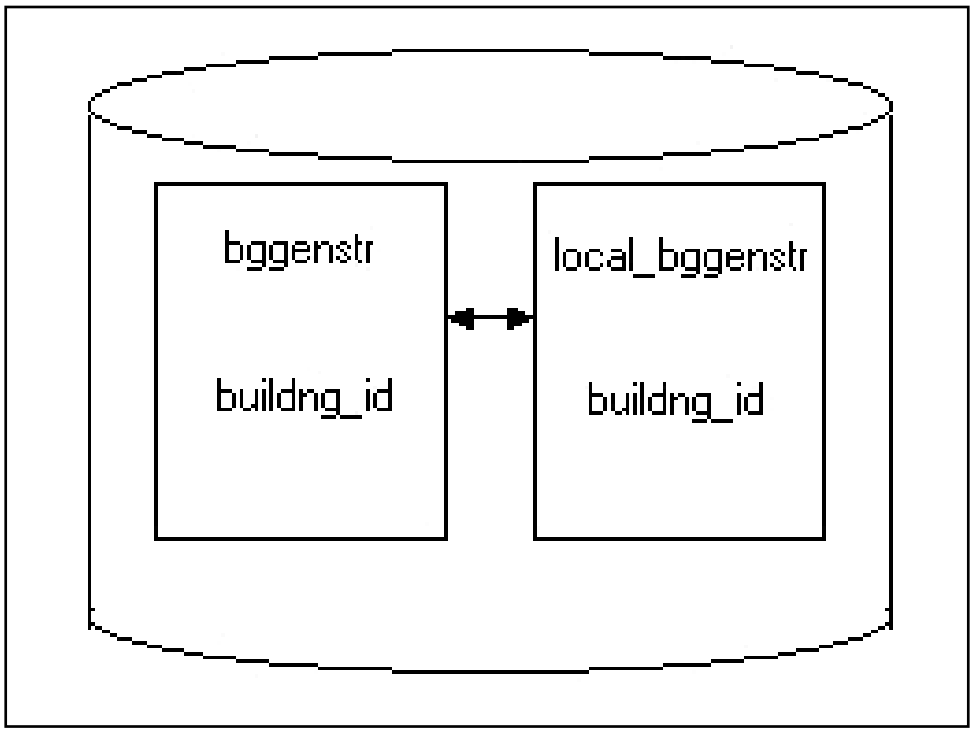

Figure F-1. Creating a local table

(4) This structure and naming convention has several advantages. First, all of the "local" tables will appear together in a tables list, and they are easily correlated to the corresponding SDSFIE/FMSFIE attribute table. This guarantees that the local attribute table will not be confused with a SDSFIE/FMSFIE attribute table, since the name exceeds the eight-character SDSFIE/FMSFIE limit and begins with the defined local identifier. By naming the first attribute as the Primary Key of the corresponding

SDSFIE/FMSFIE table, it is possible to create a join on the “buildng_id” in bggenstr with “buildng_id” in local_bggenstr, which provides the correct instance correlation.

(5) When, and if, the CADD/GIS Technology Center adds an attribute to the SDSFIE/FMSFIE that corresponds to the locally configured attribute, a single SQL query can be run that will update the values in the SDSFIE/FMSFIE table from the local table. That SQL statement would look something like:

UPDATE bggenstr INNER JOIN local_bggenstr ON bggenstr.buildng_id =
local_bggenstr.buildng_id SET bggenstr.gate_num = local_bggenstr.gate_num;

(6) The local attribute can then be deleted from the local attribute table, or the local attribute table may be deleted if that is the only local attribute in the table (see Figure F-1).

d. Domain values. Use SDSFIE/FMSFIE domain table naming conventions and definitions (short *names* of eight characters in length). (The SQL Generator and Access Builder software applications currently construct domain tables using the SDSFIE/FMSFIE short names; i.e., Domain Table Name.) Use of the entire SDSFIE/FMSFIE schema is not a requirement. A subset of the SDSFIE/FMSFIE schema may be used by selecting one of the custom filters provided with each SDSFIE/FMSFIE release; or using the SDSFIE/FMSFIE Filter Maker software application to build a custom filter based on the specific entity types required. The filters will provide a SDSFIE/FMSFIE schema containing all of the selected graphic attribute tables (i.e., those attribute tables associated with the selected entity types) as well as all linked/joined SDSFIE/FMSFIE attribute tables. All of the SDSFIE/FMSFIE domain tables associated with the selected and linked/joined attribute tables are also provided. The unneeded linked/joined SDSFIE/FMSFIE attribute tables and associated domain tables may be deleted.

e. List and range domain values. Use defined SDSFIE/FMSFIE list and range domain value naming *conventions* and definitions (short names of 16 characters in length). (The SQL Generator and Access Builder software applications currently construct domain tables using the SDSFIE/FMSFIE domain value short names.) All SDSFIE/FMSFIE-defined list and range domain values must be kept for the selected domain tables. Local list and range domain tables may be included in the database using the same procedures as defined for local attribute tables.

F-3. Experienced Level

The Computer-Aided Design and Drafting/Geographic Information Systems (CADD/GIS) Technology Center currently does not provide tools that support customer customization of tables; i.e., deleting and adding attributes to SDSFIE/FMSFIE tables.

a. Entity types and entities. Use SDSFIE entity type and entity naming conventions (i.e., the naming conventions included in the SDSFIE). The SDSFIE symbology (i.e., colors, line styles, and symbols) is recommended; however, customers have the option of using their own symbology.

b. Attribute tables. Use SDSFIE/FMSFIE attribute table naming conventions and definitions. Short names of 8 characters in length or common names of up to 50 characters in length are acceptable. Use of the entire SDSFIE/FMSFIE schema is not a requirement. A subset of the SDSFIE/FMSFIE schema may be used by selecting one of the custom filters provided with each SDSFIE/FMSFIE release; or using the SDSFIE/FMSFIE Filter Maker software application to build a custom filter based on the specific entity types required. The filters will provide a SDSFIE/FMSFIE schema containing all of the selected graphic attribute tables (i.e., those attribute tables associated with the selected entity types) as well as all linked/joined SDSFIE/FMSFIE attribute tables. The unneeded linked/joined SDSFIE/FMSFIE attribute tables may be deleted.

c. Attributes.

(1) Customers have the flexibility of adding local attributes, as well as deleting SDSFIE/FMSFIE attributes not required for the customer's GIS/FM data collection, analysis, and reporting requirements. SDSFIE/FMSFIE formatting, ordering, and naming conventions must be maintained for the SDSFIE/FMSFIE attributes retained in each attribute table. The customer cannot add local attributes that duplicate the SDSFIE/FMSFIE attributes. For the SDSFIE/FMSFIE attributes, either the short name (10 characters in length) or common name (up to 50 characters in length) may be used.

(2) The following attributes must be retained in all SDSFIE and FMSFIE attribute tables:

(a) *Primary key*. The primary key attribute is required for assigning a unique identifier to the data record and for building the relationships (joins) with similar records used by the relational database software. The attribute entitled "buildng_id" is the primary key for the table "bggenstr" (see Figure F-2 below).

(b) *Experienced level users*. Figure F-2 provides an example of a possible experienced level SDSFIE/FMSFIE implementation. It contains a total of 62 attributes. The customers represented in the figure have determined that they only need to collect and maintain data for only 23 of these SDSFIE/FMSFIE attributes, and they have added one "local" attribute entitled "my_data." Experienced level users also have the following options:

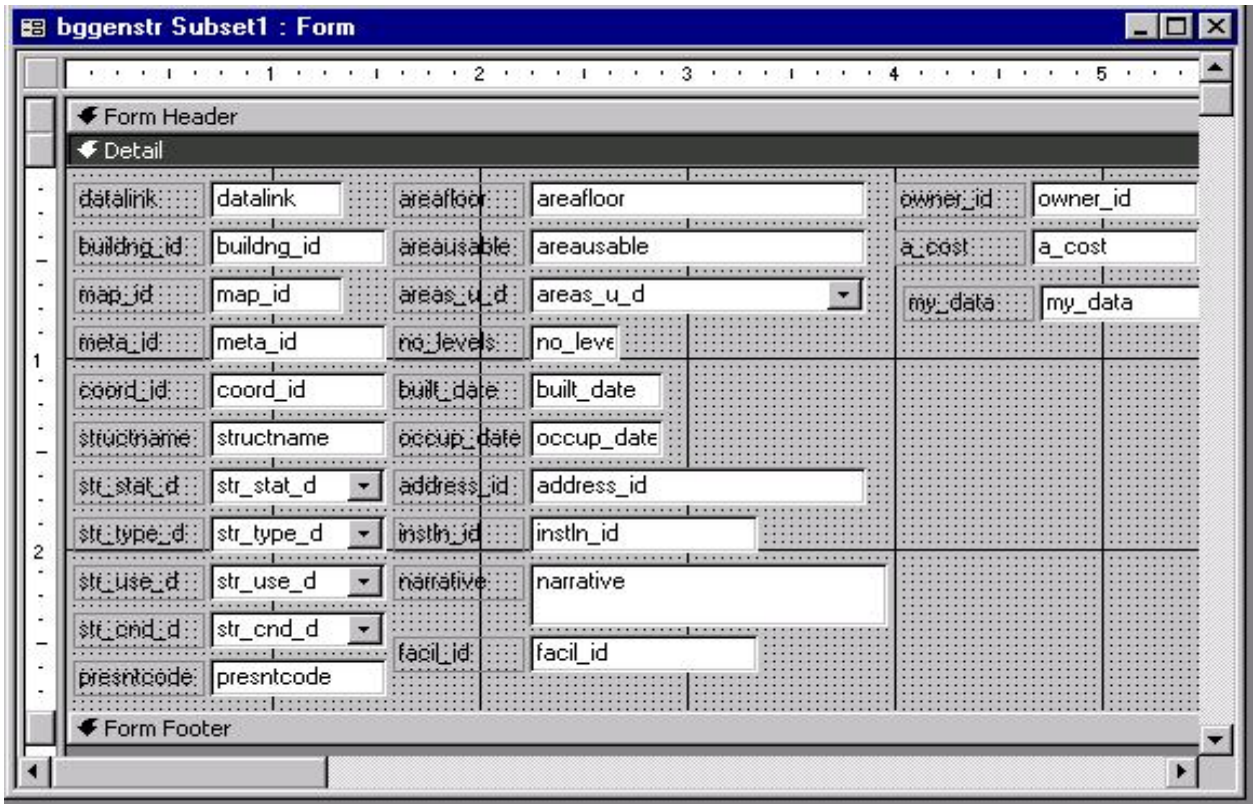

Figure F-2. Bggenstr Subset1

- The order of the attributes in a table may be changed.
- "Date/Time" data types may be used in lieu of the SDSFIE/FMSFIE integer date/time fields.
- Boolean (Yes/No) data types may be used in lieu of the SDSFIE/FMSFIE Boolean Domain Table (d_boolean).
- Shorter character fields than those specified in the SDSFIE/FMSFIE may be used.

d. Domain tables. Use defined SDSFIE/FMSFIE domain table naming conventions and definitions. Acceptable names can be either short at 10 characters or long at up to 32 characters in length. Use of the entire SDSFIE/FMSFIE data structure is not a requirement. A subset of the SDSFIE/FMSFIE schema may be used by selecting one of the custom filters provided with each SDSFIE/FMSFIE release; or using the SDSFIE/FMSFIE Filter Maker software application to build a custom filter based on the specific entity types required.

e. Domain values. Customers have the flexibility of adding local domain values and deleting SDSFIE/FMSFIE domain values not required for the customer's GIS/FM data collection, analysis, and reporting. SDSFIE/FMSFIE formatting, ordering, and naming conventions must be maintained for the SDSFIE/FMSFIE domain values retained. The customer cannot add local domain values that are a duplicate of those defined in the SDSFIE/FMSFIE domain tables. For SDSFIE/FMSFIE list domain values, either the short name with up to 16 characters or the long name with up to 32 characters may be used.

F-3. Definitions

a. Entity set.

(1) Entity sets are the highest level of the SDSFIE data model structure and represent data organized at the project level. Entity sets are broad, generalized themes containing groupings (called entity classes) of features (i.e., graphic objects called entity types that can be depicted at their actual geographic locations on a map) and related graphic attribute data (i.e., information about the feature which is stored in a database table). The SDSFIE Release 2.00 structure contains the following 26 entity sets:

- Auditory
- Boundary
- Buildings
- Cadastre
- Climate
- Common
- Communications

- Cultural
- Demographics
- Environmental Hazards
- Ecology
- Fauna
- Flora
- Future Projects
- Geodesy
- Geology
- Hydrography
- Improvements
- Landform
- Land Status
- Military Operations
- Olfactory
- Soil
- Transportation
- Utilities
- Visual

(2) The appropriate entity set name is reflected in the first two characters of each attribute table name code.

(3) For Computer-Aided Design and Drafting (CADD) (e.g., MicroStation and AutoCAD) and CADD-based Geographic Information Systems (GIS) (e.g., MGE, AutoDesk Map, and Bentley GeoGraphics), the entity set name is also represented in the first two characters of each design file name code.

b. Entity class. Entity classes comprise the next level of the hierarchical SDSFIE data model structure. Entity classes contain groupings of similar features (called entity types) and related "graphic" attribute data. Each entity class is equivalent to a separate map or drawing file. Equivalent names used by various CADD/GIS software vendors are provided in Table F-1 below.

Table F-1
Various CADD and GIS Software Names

CADD/GIS Software (Vendor)	Counterpart Name for Entity Class
MGE (Intergraph)	Category or Design file
ARCINFO (ESRI)	Workspace
MicroStation (Bentley)	Design file
AutoCAD (AutoDesk)	Drawing File

The name of an entity class is represented by a three-character code, which makes up a part of the attribute table name codes (and design/drawing file-name codes for CADD and CADD-based GIS).

a. Entity type.

(1) Each entity class contains one or more entity types. An entity type is the logical name assigned to a graphic feature (i.e., an object that can be graphically depicted on a map or drawing). Each entity type has a corresponding graphic attribute table containing specific information about the entity type.

(2) An entity type is equivalent to a "coverage" in ArcInfo and a "view" in ESRI ArcView. Entity types in MGE are grouped by features.

(3) For CADD (e.g., MicroStation and AutoCAD) and CADD-based GIS (e.g., AutoDesk Map and Bentley GeoGraphics) software, entity types represent a grouping of like cartographic (or CADD) elements (called entities) assigned to separate levels/layers.

b. Discriminators.

(1) Effective use of GIS relies on the ability of the user to adequately differentiate subtle differences in geographical features, or entities. This differentiation permits greater value in output products by selectively displaying entities based on some predefined criteria. While some differentiation is determined by the assignment of the graphical properties (e.g., color), it is often useful to expand the capability of this differentiation to display only selected entities (i.e., displaying a map of only the paved roads).

(2) Historically, the CADD user has accomplished this differentiation by assigning these different entities to different layers or levels within the drawing file. Virtually all CADD systems provide a simple capability to turn various layers on or off to allow for the display of only selected entities. This technique allows the user to "discriminate" these entity types on the basis of the layer or level. The user must understand that paved roads are to be placed on a different level from unpaved roads.

(3) Other GIS applications, including ArcInfo, do not store data with level/layer assignments, but rather organize graphic entities based on a specific entity type (road) with a "discriminator" included as a part of the attached attribute data (paved or unpaved). This allows the display of all roads without having to access multiple drawing files. The additional differentiation of using layer/level or tabular attribute, or both if desired, has been included in this version of the SDSFIE to allow for differentiation of these entity types.

(4) The individual features are logically grouped into entity types with the inclusion of a discriminator field in the corresponding attribute data. Because this discriminator has only discrete values, it is defined with respect to a domain table, which defines the values that discriminate the entities. Continuing with the previous example, the attribute table associated with entity type roads (trvehrd) now contains an attribute. This attribute defines the paved status of the road (paved_d), which refers to the domain table (d_pavstt) that contains the values "PAVED" and "UNPAVED." A user or developer can now select the roads that have the paving characteristics desired, or ignore the discriminator completely and display all roads. The technique allows for maximum flexibility in displaying only the data desired to convey the maximum amount of information.

(5) Within the standard, the inclusion of the discriminator concept requires the addition of another entity category. This additional category or grouping of entities is defined as entity types that consist of a given graphic feature "road." These entity types are normally included in the standard as nouns, while the discriminators represent adjectives that further define or describe these nouns. This modification significantly changes the format of this release of the standard.

c. Attribute table.

(1) An attribute table is a relational database table containing data, or information, about a specific SDSFIE entity. SDSFIE attribute tables are linked directly to a graphic entity (using the electronic tools provided with CADD/GIS software). They are classified as graphic (i.e., SDSFIE) attribute tables. FMSFIE attribute tables are indirectly linked to graphic entities via Foreign Key Joins to the SDSFIE attribute tables.

(2) A database can be defined as a structured collection of data items about a specific topic. A database table can be defined as a group of similar records. A database table is like a spreadsheet where the columns represent the fields, or attributes, and the rows represent the records, such that each row will be associated with a single record. A typical GIS links the graphical element, how it is displayed on the screen, with the associated record in the data table.

(3) The SDSFIE/FMSFIE has been designed for use with relational database management system (RDBMS) software. RDBMS software provides a means of managing the related data contained in one or more database tables. Examples of RDBMS software include Oracle (Oracle Corporation) and Access (Microsoft Corporation), SQLServer, and Informix. RDBMS software provides electronic tools for defining relationships (i.e., connections) between the different database tables. These relationships can be defined as one to many (most common); one to one (rare, usually merge tables to one); and many to many (needs a junction table).

(4) The name code of each attribute table is composed of eight characters, due to a design requirement to support both DOS and Windows 3.1-based GIS and database (e.g., dBase) programs. The first two characters and the next three characters in the table name reference the parent entity set and the entity class, respectively. The last three characters in the table name represent a particular attribute table.

(5) Some equivalent terms for attribute table used in GIS and relational database management system (RDBMS) software include the following:

- "feature attribute table" - in MGE.
- "attribute table" - in ArcInfo.
- "database table" - in RDBMS software.

d. Domain table.

(1) Domain tables contain standardized lists of permissible values for specific attributes. They provide a predefined finite set of allowable values, which may be enlarged by each user. Included are diverse tables of units of measure, types, styles, status, names, methods, materials, dispositions, sources, dimensions, data, classes, building numbers, etc. The user can add to these lists and range domains installation-specific values as needed.

(2) Two categories of domain tables are included in the SDSFIE: list domains, which provide a "picklist" of allowable discrete values; and range domains, which provide a range (i.e., the minimum and maximum) of allowable discrete values.

(3) Each domain table name code is restricted to eight characters with the first two characters being "d_." All attributes whose values are constrained by a domain value have a name code which ends in "_d." The name code for all attributes whose values are defined by the "unit of measure" domain table (d_uom) end in "_u_d."